

Chapter 1

Literature Review

1.1 Introduction

This chapter details the previous work done in and around the area of Fire Engineering by other academics and research staff. This section identifies the research gap that this research is intended to fill.

Fire engineering covers a broad aspect of research, from investigation into costs, materials, construction methods and the psychology of occupants during fires. This chapter covers the research done in these areas, though mainly in regards to construction and costs of fires. It also includes a brief literature review on the construction of a software tool as this was the predicted outcome of this research.

1.2 Fire Protection

Society dictates that people should be able to enjoy a reasonable level of safety during day to day activities and as such, part of this is protection from natural and man made phenomena such as fire. Such demands for safety, especially after large incidents where a considerable loss of life or property occurred, led to the initial development of the fire building regulations (Stollard and Johnston, 1994). These building regulations developed over time into the current form we have today in the UK (Communities and Local Government, 2006). These regulations are in place so that the building can be designed safely to allow occupants to escape should a fire occur and to prevent excessively quick fire spread, again, allowing for escape. This focus on life safety is the main concern behind the current regulations and appears to have been reasonably successful, as fire deaths have declined steadily over the past decade (Department for Communities and Local Government, 2010). However, there has been a steady increase in the cost of fires over the same period (Association of British Insurers, 2009). These figures seem to show that whilst the building regulations seem to be working for reducing and keeping the number of fire deaths low, they don't seem to be any reduction to the cost of fires by constructing to meet the building regulations.

Whilst Approved Document B lays down the building regulations in relation to fire safety and details the easiest method of meeting these regulations, it is by no means the only method of meeting the regulations. Designers also have access to two British Standards that set out methods of meeting the regulations. These British Standards are more advanced methods of designing buildings and therefore are used in the larger and more complex buildings. These two codes are also more open in how the design of the building can attain fire safety certification. BS 9999: Code of Practice for Fire Safety in the Design Management and Use of Buildings (BSI, 2008) is a relatively new code that attempts to make it easier for designers to

incorporate fire safety into more complex structures without having to use PD 7974: Application of Fire Safety Engineering Principles to the Design of Buildings (BSI, 2003) which is used in the most complex of buildings because it reduces fire safety to the first principles of fire science and requires an understanding of flame spread, fire dynamics and combustion science to use effectively.

To meet the fire regulations, buildings have to be protected by differentiating fire protection systems. These systems delay the spread of fire and make it easier for the occupants to escape. The design of the building itself and the restriction in the use of combustible materials is one method of preventing fire spread is the main aspect behind the regulations. This method of preventing fire spread is referred to as passive fire protection, as the protection does not need to change state to complete its function in preventing fire spread. However, additional systems are installed, such as fire alarms and extinction systems are installed and these are active fire protection measures. The combination of both active and passive measures provides the complete fire protection design for a building. Previous research has suggested that the inclusion of active measures such as sprinklers can reduce the passive protection measures included in a building (Baldwin and Thomas, 1974) though it's still believed that both systems should be used complementarily (Haack, 2004).

There has been much work on the use of active protection measures in buildings, especially sprinklers (Hall, 2010, Melinek, 1993a, Vaidogas and Šakenaite, 2011, Melinek, 1993b) and investigating the costs of sprinklers in various buildings, such as in car parks (Li and Spearpoint, 2004), residential properties (Butry, 2009)

1.3 Decision Support Systems

A Decision Support System (DSS) can be defined as a “computer based systems that help decision makers confront ill-structured problems through direct interaction with data and analysis models” (Sprague and Watson, 1993). These systems allow designers and managers to access help in the decision making process.

The majority of DSS tool have similar features in common. These are:-

- They are non routine and involve frequent *ad hoc* analysis, fast access to data and generation of non standard reports
- They often address “What if?” questions
- They have no obvious correct answer; a manager has to make qualitative trade offs and take into account situational factors

These features are discussed in Value Analysis: Justifying Decision Support Systems (Keen, 1981). This paper discusses the fact that DSS tools are helpful in the role they play but are difficult to quantify in regards to cost savings. However, Keen describes that tools should be:-

1. Flexible
2. Easy to use
3. Responsive
4. Communicative

He explains that it should be flexible to handle different situations, easy to use so it can be put into different users workflow, responsive as it must not impose a structure on the user and be rapid in its calculations and

communicative so that the end user can get the knowledge they need from the tool quickly and easily. By following these four “rules”, the tool should be of the most use to the end users.

In his book (Keen, 1980), Keen describes the perceived benefits of a DSS system. These are:

1. Increase in the number of alternatives examined
2. Better understanding of the business
3. Fast response to unexpected situations
4. Ability to carry out *ad hoc* analysis
5. New insights and learning
6. Improved communication
7. Control
8. Cost savings
9. Better decisions
10. More effective teamwork
11. Time savings
12. Making better use of data resources

Not all of these 12 benefits can be applied to the design tool in this research, however the proposed tool does achieve a number of these points, specifically points one (alternative approaches), eight (cost savings), nine (better decisions), eleven (time savings) and twelve (Making better use of data resources). By allowing the end user to run more alternatives to the fire design, it is hoped that a better, more cost effective solution can be put forward as an alternative to the other plans a designer might submit.

1.4 Data Collection

Most of the data for this PhD is incident data and has already been collected by the Fire and Rescue Service (FRS) for Department of Communities and Local Government (CLG) use as well as the data from the Fire Protection Association (FPA). However, data needs to be collected from fire engineers and those involved within the building industry to investigate the current practise and how the fire protection is specified in a building, as based on literature, the mindset appears to be that the structural engineer designs the structure and the architect and fire engineer design the fire protection (Alfawakhiri and Carter, 2005). Collecting data will allow the DSS to focus on those that are responsible for the design of the fire protection within a building and therefore the tool can be customised to the needs of that intended group.

1.5 Software Design

As the outcome of the research is a proposed methodology for a DSS system, then a brief literature review should cover the design of the software. The tool is designed to be used by fire engineers and possibly fire engineering clients, so the tool will cater to the needs of the fire engineers. The definition of tool in this research follows that given by Lockley and Sun (Lockley and Sun, 1995) which is “a computer program that is used by engineers to perform analysis of a building or its services (prior to relisation) for the purpose of making, modifying or evaluating design decisions”.

1.5.1 User Interface

As discussed in 1.3, Keen puts forward the areas a DSS should follow to allow the best use. These were that it should be flexible, easy to use, responsive and communicative. These aspects are met in the coding of the software. For example, ease of use comes down to the design of the Graphical User Interface (GUI). A well developed GUI will allow users to navigate the program intuitively. In the design of GUI's, Shneiderman and Plaisant lay down 8 "golden rules" of interface design. These are:-

1. **Strive for consistency** - Program design should be consistent throughout, such as layout, fonts, design and where possible, user actions and terminology.
2. **Cater to universal usability** - Recognise the needs of the users and design accordingly. Different users will use the tool differently (further discussed in a paper by Udema (Uduma and Morrison, 2007) and also mentioned by Sprague in his paper setting out the framework for DSS creation (Sprague, 1980).) which stated that different skill levels of users would use the tool differently, depending on the users level of experience in the field the tool is designed for). This means adding help for novice users and shortcut keys and faster pacing for more experienced users.
3. **Offer informative feedback** - For each user action, the software should provide feedback though the feedback should follow the scale of actions (minor feedback for minor actions, major feedback for major actions)
4. **Design dialogs to yield closure** - Sequences of actions should be organised into groups and should have a beginning, middle and end. Feedback should be given at the completion of a set of actions so the user knows the item is complete (for example, e-commerce sites show a checkout completion screen to let users know this set of actions has been completed)
5. **Prevent errors** - Design the program to prevent errors, such as only allow numbers to be entered into a field that only needs numerical data entered. If an erroneous value is entered, provide feedback to the user and let them correct it. Allow them to only have to correct the erroneous value rather than redo the entire form.
6. **Permit easy reversal of actions** - As much as possible, allow for easy reversal of errors. This allows users a sense of relief, knowing they can undo any error - this allows for exploration of unfamiliar options.
7. **Support internal locus of control** - Experienced operators desire the sense that they are in command of the interface and the interface should be designed accordingly. Surprising interface actions, tedious sequences of data entries or the inability to gather information will build dissatisfaction with the product.
8. **Reduce short term memory load** - Human short term memory means that displays should be kept simple and short. Where appropriate, online access should be provided to command-syntax, abbreviations, shortcuts and other information.

These rules will lead to the optimised GUI design that all parties, regardless of experience can use.

1.5.2 Software Methodologies

As DSS tools are software programs, they can be designed following any of the current software design methodologies. A design methodology is the method of software creation, following a set system.

Design methodologies often fall into one of two categories, Sequential or Iterative methodologies (Whitten *et al.*, 2003). Figure 1.1 demonstrates a sequential methodology. All development steps in a stage are completed before progressing onto the next step. This kind of methodology results in a longer time before the end user gets to use the software as all steps are completed before the final product is released.

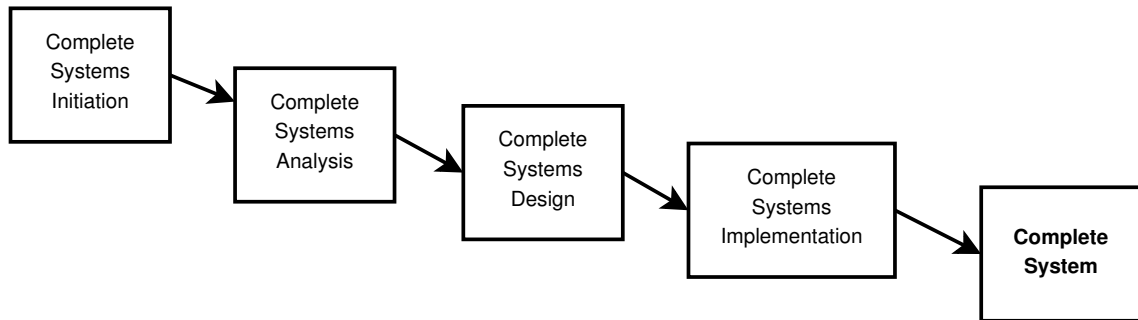


Figure 1.1: Sequential/Waterfall Methodology

Figure 1.2 shows an iterative step methodology. This methodology means that the end user can see progress on software throughout the design of the software as he will be involved in the testing of each partial system produced. This allows for more constant communication between the end users and the software programmers, meaning that the end product is likely to be more thoroughly tested than a sequentially designed piece of software.

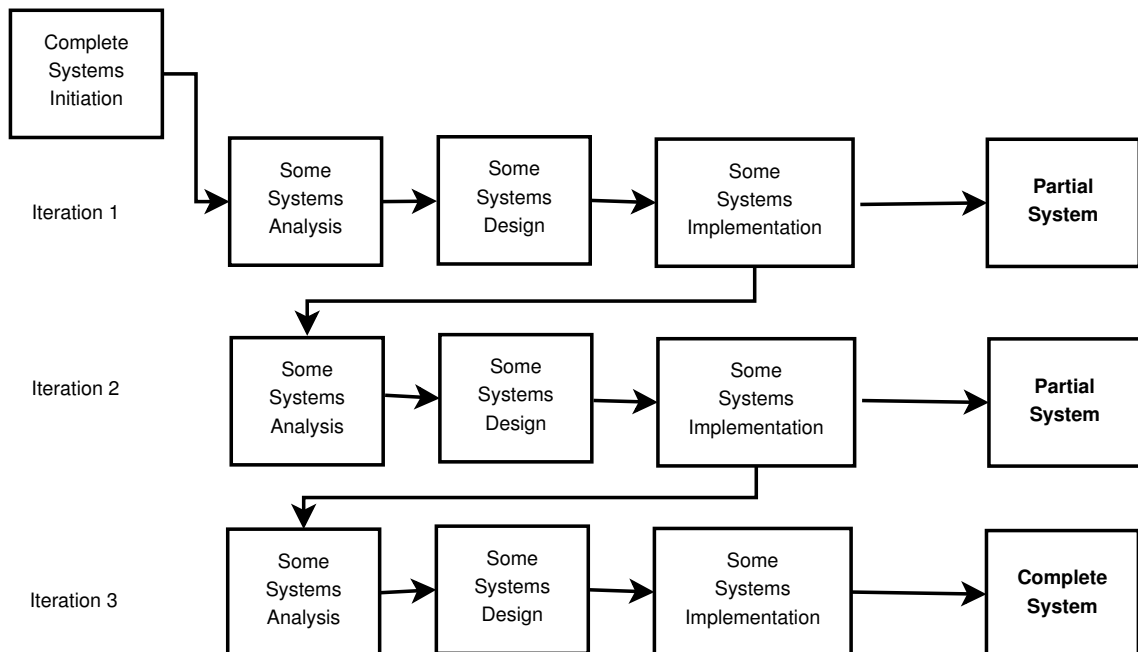


Figure 1.2: Iterative/Incremental Methodology

RAD

Rapid Application Development or RAD, is an iterative design methodology as its name implies, for the rapid development of software. It focuses on getting the end users interacting with the program during

the development so that each cycle or iteration can be tested. The principle behind the beta versions and prototyping is that users will have a better idea of what they want when they see part of the program already working. RAD focuses on a few main points.

- Emphasis is on reducing time therefore phases of design and programming are consolidated and accelerated.
- In each iteration, only some design specifications will be considered.
- Assumption is that errors will be corrected in the next iteration.
- After each iteration, end users are invited to test the software.
- Based on feedback, designers will make changes until a version is deemed worthy of implementation.

The design phase within a RAD methodology is very short - the majority of design is done during each iteration.

Agile

In 2002, a publication by VTT Publications stated that agile software development was an umbrella term for various different software methodologies (Abrahamsson *et al.*, 2002). All of them share the same principles of quick iterative design, similar to the RAD methodology described above. In 2001, an manifesto was setup to document what an Agile methodology should consist of. This can be found at the Agile manifesto page (Beck *et al.*, 2001). This states the following principles:

1. Individuals and interaction over process and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

These principles mean the various methodologies that fall under the Agile umbrella all share common roots.

Of the Agile methods, there are a few which stand out as being more common than other methodologies (Cohen *et al.*, 2003). One of these is XP or Extreme Programming, put forward by Beck (Beck, 2000) and the other is Scrum, first described in 1986 (Takeuchi and Nonaka, 1986).

XP

Extreme programming is focussed on small team developments, with 2-10 people involved in the design team. XP seeks to prevent schedule slips by having short release cycles on software (Beck, 2000). However, due to the nature of the methodology, of inter team communication and the need for co-location of the design teams. It also has a very short iteration time of about 2 to 4 weeks.

Scrum

Scrum is another team driven methodology that relies on an iterative, incremental steps. Scrum relies on:

- Transparency - Ensures that all aspects that affect the outcome are visible to those managing the work.
- Inspection - Work must be inspected frequently.

- Adaption - If an inspection determines something is outside acceptable limits, the inspector will adjust the process.

This list was taken from (Schwaber and Sutherland, 2010). The roles in a Scrum are those of the team (the programmers) and the ScrumMaster. The team are designed to optimise flexibility and productivity. Iterations are called sprints and these sprints are time boxed (done within a set time limit). During the sprint, the ScrumMaster makes sure that no changes are made that would affect the Sprint Goal. As the Sprints progress, the iterations come together to form the final product.

Bibliography

- Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. *Agile Software Development Methods*. VTT Finland, 2002.
- Alfawakhiri, F. and Carter, C. J. Selecting the “Right” Fire Resistant Design. *Practice Periodical On Structural Design*, 10(1), 3 – 7, 2005.
- Association of British Insurers. *Tackling Fire: A Call For Action*. Association of British Insurers, 2009.
- Baldwin, R. and Thomas, P. H. Passive and Active Fire Protection - The Optimum Combination. *Fire Technology*, 10(2), 140–146, 1974.
- Beck, K. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2000.
- Beck, K., Cockburn, A., Jeffries, R., and Highsmith, J. Agile manifesto. Online, 2001.
- BSI. PD 7974: Application of Fire Safety Engineering Principles to the Design of Buildings. 2003.
- BSI. BS 9999: Code of Practice for Fire Safety in the Design Management and Use of Buildings. 2008.
- Butry, D. T. Economic Performance of Residential Fire Sprinkler Systems. *Fire Technology*, 45(1), 117–143, 2009.
- Cohen, D., Lindvall, M., and Costa, P. *Agile Software Development*. Data and Analysis Center for Software, 2003.
- Communities and Local Government. *Approved Document B: Fire Safety*. NBS, 2006 edition, 2006.
- Department for Communities and Local Government. *Fire Statistics, United Kingdom 2008*. Department for Communities and Local Government, 2010.
- Haack, A. Passive Fire Protection - Sufficient or Complimentary to Active Systems? In *International Symposium on Tunnel Safety and Security*. STUVA and ITA, 2004.
- Hall, J. R. *U.S. Experience With Sprinklers And Other Automatic Fire Extinguishing Equipment*. National Fire Protection Association, 2010.
- Keen, P. G. W. *Decision Support Systems And Managerial Productivity Analysis*. Cambridge, Mass, 1980.
- Keen, P. G. W. Value Analysis: Justifying Decision Support Systems. *MIS Quarterly*, 5(1), 1 – 15, 1981.
- Li, Y. and Spearpoint, M. Cost-Benefit Analysis of Sprinklers For Property Protection In New Zealand Parking Buildings. *Journal of Applied Fire Science*, 12(3), 223 – 243, 2004.
- Lockley, S. R. and Sun, M. The Development of a Design Tool. *Pattern Recognition*, 28(10), 1499 – 1506, 1995.

- Melinek, S. Effectiveness of Sprinklers in Reducing Fire Severity. *Fire Safety Journal*, 21(4), 299 – 311, 1993a.
- Melinek, S. J. Potential Value of Sprinklers in Reducing Fire Casualties. *Fire Safety Journal*, 20(3), 275–287, 1993b.
- Schwaber, K. and Sutherland, J. *Scrum Guide*. Scrum.org, 2010.
- Sprague, J., Ralph H. A Framework For The Development Of Decision Support Systems. *MIS Quarterly*, 4(4), 1 – 26, 1980.
- Sprague, R. H. and Watson, H. J., editors. *Decision Support Systems: Putting Theory into Practice*. Prentice Hall, 3rd edition, 1993.
- Stollard, P. and Johnston, L., editors. *Design Against Fire : An Introduction to Fire Safety Engineering Design*. E & FN Spon, 1994.
- Takeuchi, H. and Nonaka, I. The New New Product Development Game. *Harvard Business Review*, 64(1), 137 – 146, 1986.
- Uduma, L. and Morrison, G. R. How do Instructional Designers Use Automated Instructional Design Tool? *Computers in Human Behavior*, 23, 536 – 553, 2007.
- Vaidogas, E. R. and Šakenaite, J. A Brief Look at Data on the Reliability of Sprinklers Used in Conventional Buildings. *Journal of Civil Engineering and Management*, 17(1), 115 – 125, 2011.
- Whitten, J. L., Bentley, L. D., and Dittman, K. C. *Systems Analysis and Design Methods*. Irwin/McGraw-Hill, 6 edition, 2003.